COPY

Naval Research Laboratory

Wash ngton, DC 20375-5000

NRL Memorandum Report 6691

AD-A224 700

An Algorithm for Improved Gating Combinatorics in Multiple-Target Tracking

M. R. ZUNIGA, J. M. PICONE, AND J. K. UHLMANN

Center for Computational Physics Development
Laboratory for Computational Physics and Fluid Dynamics Division

August 2, 1990

DTIC
ELECTE
AUG 0 3 1990
D

Approved for public release; distribution unlimited.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (*Leave blank*)

2. REPORT DATE
1990 August 2

3. REPORT TYPE AND DATES COVERED
Interim

4. TITLE AND SUBTITLE
An Algorithm for Improved Gating Combinatorics in Multiple-Target Tracking

5. FUNDING NUMBERS
PE - 63223C
PR - 44-3287-00
WU - DN159-68

6. AUTHOR(S)
M. R. Zuniga, J. M. Picone and J. K. Uhlmann

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Research Laboratory
Washington, DC 20375-5000

8. PERFORMING ORGANIZATION REPORT NUMBER
NRL Memorandum Report 6691

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Office of Naval P earch/SDIO
800 N. Quincy St t
Arlington, VA 22217-5999

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (*Maximum 200 words*)

This paper describes an efficient approach to gating. It is shown that the overall algorithm scales significantly better than quardratic, even when reports have unequal timestamps within a scan.

14. SUBJECT TERMS
Multiple - target tracking, Gating
Near-neighbor algorithms

15. NUMBER OF PAGES
22

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev 2-89)
Prescribed by ANSI Std Z39-18
298-102

CONTENTS

Accession For
NTIS GRA&I
DTIC TAB
Unannounced
Justification
By
Distribution/
Availability Codes
Avail and/or
Dist Special
A-1

AN ALGORITHM FOR IMPROVED GATING COMBINATORICS IN MULTIPLE-TARGET TRACKING

I. Introduction

In this paper we describe a method for significantly reducing the computational complexity required for observation-track gating in multiple target tracking. We define the gating process as follows: given a set of N_R observations and N_T tracks, identify all observation-track pairs whose scores fall above a chosen threshold. The score for observation-track pair (i, j) is defined as the function:

$$S_{ij}(dX_{ij}, \Gamma_j) = \frac{\exp(-dX_{ij}^t \Gamma_j^{-1} dX_{ij}/2)}{(2\pi)^{\frac{d}{2}} \sqrt{|\Gamma_j|}}, \tag{1}$$

where d is the measurement dimension, Γ_j is the residual covariance matrix of the track j, and dX_{ij} is the residual vector of the pair (i, j), and these arguments are taken to be valid at the same time. In addition, throughout this study we shall use the score threshold, S_j, chosen for every observation j such that:

$$S_{ij} \leq S_j. \tag{2}$$

In part because Eq. (1) is very expensive to evaluate, much of the previous work on gating has emphasized the use of intermediate or "coarse" gating criteria which replaces the calculation of Eq. (1) with a function that is computationally cheaper to evaluate. The result is the identification of a superset of candidate pairs which includes the pairs that satisfy Eq. (2). We denote the subset satisfying Eq. (2) as either the pairs which correlate at gating or the pairs which satisfy the final gate. Typically this pre-processing includes defining a gating volume, V_G, based on the numerator of the right-hand side (RHS) of Eq. (1) so that a coarse correlation measure is evaluated assuming a Gaussian

Manuscript approved June 4, 1990.

distribution as in Eq. (1). For example, the pairs pass one gate if $\gamma V_G < (dX^t \Gamma_j^{-1} dX)$, where γV_G is a threshold that may be obtained from a table or an error function integration for a certain probability of correlation [1, page 97] [2, pages 88ff]. These pairs might also be pre-processed by coarser gating criteria with larger V_G but which are cheaper to evaluate [2, page 91], [1, page 97]. Ideally, one completes the gating calculation by computing S_{ij} for the set of candidate pairs and performing the comparison in Eq. (2). Overall processing work would be reduced since only the pairs with sufficiently high coarse correlation values would be re-evaluated using the numerically expensive function of Eq. (1).

These techniques address the coefficient of the scaling but not the scaling itself since they explicitly apply a coarse correlation function to all $N_T N_R$ possible pairs. If the number of pairs explicitly evaluated by coarse gating are of order $(N_T N_R)$, then for sufficiently large N_T and N_R, real-time processing can be precluded on any computer even with the use of numerically simple coarse correlation functions. More recently Nagarajan et al.[1, pages 97-98] have recognized that one can can reduce the number of times that a coarse correlation function is evaluated. An algorithm is presented for doing so, but the analysis of scaling is incomplete. In particular, neither cost nor scaling equations of the gating process are presented in terms of parameters important in multitarget tracking, e.g. N_T or N_R. Partitional spatial clustering methods have been proposed to efficiently perform gating by exploiting features of the target distribution[3][1]. However, improved scaling cannot be guaranteed for these data dependent algorithms. The objective of this paper is to present an efficient gating algorithm and to analyse its scaling. We shall show that the overall algorithm scales significantly better than quadratic even when reports have unequal timestamps within a scan. Our algorithm is compatible with virtually all of the previous work on auxiliary gating criteria and coarse gates. As a secondary objective we present an inexpensive algorithm for calculating a coarse time independent gate volume for the criterion of Eq (2).

II.Preliminary Details

By a d-dimensional report, or observation, we mean a set of d elements measured simultaneously at some specified time. We call this time the validity time of the observation or the *timestamp* of the observation. We require that the timestamps fall within a period of time called a *scan* of length τ, where the times of the reception of the first and the last observations fix the beginning and the end of the scan. A track is an estimate that in some sense converges to the "true trajectory" as the number of observations correctly matched to the track increases. There are ℓ position components to each track at any one time. Similarly, the observations are ℓ-dimensional in position, with $\ell \leq d$.

In our analysis and numerical simulations we employ algorithms which find near neighbors of points in ℓ−dimensional position space, where nearness is defined by the Euclidean metric. They are used, for example, to find the tracks with mean positions near a given observation position. Essential is that: (1) these algorithms find *all* the neighbors in some expected optimal or near optimal time, and (2), that their performance be relatively insensitive to spatial distributions. After examining several search algorithms,

[1] For cluster type distinctions, see [4]

we selected a BLD-enhanced k-d search tree for our tests [5][6]. These data structures are known to have a worst case single-query search time scaling of $O(N^{1-\frac{1}{k}} + N_G)$, where N is the size of the database being searched and N_G is the number of near neighbors returned. In many applications (including our tests) their average case search time scaling approaches $O(\log N + N_G)$. For a background on multidimensional search structures the reader might consult Bentley [7], Friedman et al. [8] and [9], Hanan [10] and Mehlhorn[11]. For an introduction to these algorithms the reader might consult [12].

III. Problem Overview in Light of Combinatorics

Given a set of N_T tracks and a set of N_R reports, there are at most $N_T N_R$ scores S_{ij} which can be formed. Of these, a fraction q of them will fall above the thresholds and satisfy Eq. (2), where q could be as low as $1/N_T$ or $1/N_R$ or smaller. Ideally we would only calculate the $qN_T N_R$ scores; at worst we would calculate all $N_T N_R$ scores. An example of a quadratic scaling approach, for the case when reports have unequal timestamps, is the following technique: integrate the equations of motion of each of the N_T tracks to the times of each of the N_R reports and compute the scores. For each report, keep those scores that are above the desired threshold. The dominant cost of this is the $O(N_R N_T)$ score calculations and integrations. Of course, if each score calculation is replaced by a coarse gate calculation, the scaling is still quadratic.

Intuitively, when tracks and reports are at the same time, tracks and reports that are close in position tend to be correlated. Of course, Eq. (1) specifies the meaning of correlation at gating and shows that other parameters in addition to mean positions must be considered. The covariances will in part determine a gate volume around the mean positions, and we conceptually relate gating to geometry by saying that reports and tracks which gate with each other are those pairs with intersecting gate volumes. Let N_n be the number of tracks per report that should gate, as determined by Eqs. (1) and (2). Let the gating volumes be determined ideally in the sense that the set of pairs which should gate by Eqs. (1) and (2) is identical to the set of pairs which gate. Let ρ be the object density and V_{IG} the ideal gating volume per report. Let the average of a quantity X over all the reports be given by $\overline{X}$. Then the total number of gating pairs is $\overline{N}_n N_R = \overline{\rho V}_{IG} N_R = qN_T N_R$.

The prescription herein for calculating "not much more" than the required $qN_T N_R$ scores involves in part using estimates of V_{IG}, say V_G, in a search structure for identifying the pairs. We have chosen a neighborhood search volume to be spherical in nature, although it is not necessarily optimal. A search radius R_G specifies the search volume V_G. We denote R_G by R_0 when a given report has the same timestamp as the track file to be searched. When the number of correlations that should be made is small, i.e. when $qN_T N_R$ is not comparable to $N_T N_R$, then $\overline{\rho V}_{IG} N_R$ is also small. Assume we can find an R_0 per report such that: (1) the actual search neighborhood per report, V_G, includes V_{IG} and (2) $\overline{\rho V}_G$ is comparable to $\overline{\rho V}_{IG}$.

When there is a distribution in time of tracks and reports throughout a scan, then the required search radius R_G might on average define a search neighborhood so much larger than V_{IG} such that the number of candidate pairs found is no longer comparable to $qN_T N_R$. After all, we cannot merely superimpose the tracks and reports and ask which error ellipses intersect since evaluation of Eq. (1) requires that the function arguments correspond to the same time. If we instead ask which ellipses would intersect *if* they were at the same time, then the gating volume around the report's position should take into account bounds on the location possibilities of the objects due to their dynamics and the time differences between the report and the tracks. In this case the estimate of the gate volume is also time dependent, i.e. $V_G = V_G(\delta T, R_0)$, and we model its search radius as

$$R_G = R_0 + \alpha|\delta T| \;, \tag{3}$$

where α is some upper bound on the velocity and δT is the maximum time difference possible between any track and a report within the scan τ and possibly equal to τ itself. Thus scaling could depend on the two parameters on the RHS pf Eq. (3). In discussing our algorithm we find that two limiting cases of Eq. (3) are particularly convenient:

$$\overline{\alpha|\delta T|} \;\; << \overline{R}_0, \tag{4a}$$

and

$$\overline{\alpha|\delta T|} \;\; >> \overline{R}_0. \tag{4b}$$

We denote the former case as the "small scan length search" case and the latter case the "large scan length search" case. Because of Eq. (3) and Eq. (4a), the conclusions of the next section on the zero scan length case apply to the small scan length case. The following section introduces the general algorithm through a discussion of the large scan length case. The mathematical analysis of the cost for any value of $\overline{\alpha|\delta T|}/\overline{R}_0$ is presented afterwards. We also prove that the general algorithm has its worst case scaling in terms of α and τ when Eq. (4b) holds.

IV. Zero Scan Length

In the idealized case of the zero scan length, all reports from a given scan have identical timestamps. To perform gating, the track file is projected to the time of the reports, then the search radius will not depend on object dynamics since $|\delta T| = 0$ in Eq. (3). For the solution of the equal timestamp case, therefore, we calculate a search radius from a given score threshold. More specifically, in doing a search per scan per report j on a track file, we will determine the following: given a score threshold S_j, calculate the radius per report, $R_G = R_0 = R_0(S_j)$, such that all the report-track pairs which are separated by distances larger than R_0 will *not* have scores above the threshold. Calculation of this radius is complicated by the fact that the tracks in our database have nonidentical covariance matrices, or hyperellipses, of nontrivial distribution in volume and shape. Consider, however, the case where $d = 1$. For a one dimensional Gaussian function with a fixed Γ, the function value is strictly nonincreasing as its argument R^2 increases and falls below the threshold after R^2 gets larger than some value, say R_0^2. One can then determine analytically an R_0^2 that is independent of the Γ distribution. Figure 1 below is a contour plot for the function $S(\gamma, r) = \exp(-r^2/2\gamma)/\sqrt{(2\pi\gamma)}$. The lines in the figure are actually iso-score lines. Notice that each exhibits a maximum radius r_{max} such that for scores exceeding the threshold, all possible r are smaller than r_{max}. This r_{max} is a suitable choice for R_0. Appendix I has a derivation for a useful search radius for dimension $d \geq 1$; the result is:

$$|\delta\mathbf{R}|^2 \leq \sum\left(\frac{\delta X_k}{\alpha_k}\right)^2 \leq \frac{M}{2\pi e\langle\alpha_k^2\rangle}(S_j)^{-\frac{2}{d}} \equiv R_0^2 \quad . \qquad (A12,5)$$

where $\langle\alpha_k^2\rangle$ is an upper bound on the maximum eccentricity among the error ellipses in a scan.

Equation (A12) gives a suitable choice for R_0 since: (1) all tracks which *can* correlate with observation j will be within R_0 of the observation, and (2) it is a track-independent radius and thus can be used for a search on a track file. We should note that other prescriptions for calculating time independent gate volumes might meet the above criteria. Such prescriptions might differ in how close $V_G(R_0)$ approaches V_{GI} and how expensive it will be to calculate $V_G(R_0)$. Preliminary analysis indicates that our $V_G(R_0)$ is a rather coarse gate that is inexpensive to calculate.

Though the equal timestamp behavior of the search algorithm is straightforward enough, we include Plot 1 with data from a simulation (see Appendix II). Plot 1 verifies the search time scaling for an unequal timestamp case which is not quite the $\tau = 0$ limit, but with relatively small differences in timestamp. In this case τ and our overestimate of V_{IG} were small enough so that the dominant term in the scaling of the BLD tree is seen to be close to the $O(N_R \log N_T)$ diagonal line in the figure. The $O(N_T \log N_T)$ CPU expense of creating the data structure was also verified.

V. The General Algorithm and Analysis for the $\overline{\alpha|\delta T|} >> \overline{R}_0$ Case

When the reports have unequal timestamps what are the best times to which to integrate the tracks for making the potential pair matching? Following the textbook procedure of making all the tracks valid at the beginning or all valid at the end of the scan can result in the case of Eq. (4b) and possibly in a combinatorial bottleneck. We have made the following observations: (1) some search structures are cheap to make, CPU wise, and (2) by making various copies of the track data structure - the various copies valid at different times - a tradeoff can be made between the time spent on the creation of the data structures and the time spent on searching and scoring. We shall see that the payoff from spending more time in creating data structures increases as the scan length contribution to $\overline{V}_G$ increases, given that the other relevant parameters are held fixed.

If we integrate in time to make M_D sequential track data structures (TDS) valid at M_D equally spaced times within the scan of length τ (See Fig. 2), then any report would be at most $|\delta T| = \tau/(2M_D)$ time units away from a TDS. The average radius for the search is then decreased by the factor M_D as compared to the case where we have one TDS copy at the *middle* of the scan. Therefore the volume extent as well as the average number of candidates returned ($\overline{N}_G$) is smaller by $(1/M_D)^l$ in the isotropic dense limit l-dimensional case. More precisely, assume that the density ρ of objects in space is constant and uniform. Then the average number of candidate tracks found for each report depends on the average search volume $V_G = \gamma(\ell)\langle R_G^\ell\rangle$, where $\gamma(\ell)$ is a geometric factor depending on the dimension ℓ of the report state vector. The brackets $\langle\cdot\rangle$ denote the average over the temporal range $(t_i - \delta T,\ t_i + \delta T)$, where $i(R)$ labels the TDS selected to be closest to the report. Assuming that the time distribution of reports within the scan interval is uniform,[2] using Eq. (3) gives us:

$$V_G = \frac{\gamma(\ell)}{(\ell+1)\alpha\delta T}\Big[(R_0+\alpha|\delta T|)^{\ell+1} - (R_0)^{\ell+1}\Big], \tag{6a}$$

$$V_G \approx \frac{\gamma(\ell)}{\ell+1}\Big(\alpha|\delta T|\Big)^{\ell} \ . \tag{6b}$$

$$N_G \approx \rho\frac{\gamma(\ell)}{\ell+1}(\alpha|\delta T|)^{\ell} = \rho\frac{\gamma(\ell)}{\ell+1}(\frac{\alpha\tau}{2})^{\ell}M_D^{-\ell} \ . \tag{7}$$

The RHS of (6b) assumes the high scan length search condition of Eq. (4b), i.e.

$$\alpha|\delta T| = \alpha\tau/2M_D \gg R_0. \tag{8}$$

Since the searching time and the scoring time depend on N_G, the scoring time being directly proportional to N_G, the use of multiple extrapolated track files ($M_D > 1$) to cover the scan interval can reduce the cost of the gating process by reducing the search volume.

Plot 2 shows an example of poor scaling due to a "large" scan length. The trajectories which generated the data for Plot 2 are equivalent to those for Plot 1 except that the scan length is increased by a factor of five and therefore the average search radius is also increased. In this case the search algorithm's visible CPU cost was not $O(N_R \log N_T)$. For $N_T = N_R = 32K$, the search time increased by a factor 5, and the number of near neighbors returned was increased by a factor of about 10. Plot 3 shows how the scaling of the search time behaves for the same report scenario as in Plot 2, except that now $M_D = 5$ for the datasets of $4K$ and $32K$ tracks. The best fit lines approach the $\log N_T$ scaling line more closely. For the same runs and $N_T = N_R = 32K$, the number of near neighbors returned, and therefore the number of pairs which would be evaluated with Eq. (1), decreased by an average factor of 9.8 when M_D was changed from 1 to 5. This reduction also signifies that the number of tracks per report found to meet the time dependent gate volume was nearly reduced to its simulation optimum average value of 1. This factor of 9.8 was found to be 62.7 if the one TDS was placed at the time of the beginning of the scan instead of the recommended placement for $M_D = 1$!

Was the price paid for overhead of the multiple data copies too high so that the overall cost would still scale poorly? Let $\overline{N}_G$ be the report averaged number of near neighbors returned. Then the total CPU cost can be modeled as

$$C(N_T, N_R, M_D, \overline{N}_G) = C_e M_D N_T + C_d M_D N_T \log N_T + C_{se} N_R(\log N_T + \overline{N}_G) + C_{sc}\overline{N}_G N_R \tag{9}$$

where the terms on the RHS of Eq. (9) give, respectively, the cost for integrating the tracks to the desired time of the data structures, the cost of making the tree data

[2] An overestimate of the worst case is when the reports are $\delta T = \tau/(2M_D)$ time units away from a TDS, i.e. at the furthest possible time difference, where the reported average case is small by $2^{\ell}/(\ell+1)$

structures, the cost of searching the appropriate tree data structure for each report, and the cost of scoring the pairs.

Equation (9) with our model for Eq. (7) has a minimum value which occurs for the following optimal M_D

$$M_{D0} = (\ell\frac{\kappa_2}{\kappa_1})^{1/(\ell+1)}, \tag{10}$$

where

$$\kappa_1 \equiv N_T(C_e + C_d \log N_T),$$

$$\kappa_2 \equiv N_R(C_{sc} + C_{se})\frac{\gamma(\ell)}{\ell+1}\rho(\frac{\alpha\tau}{2})^{\ell} \quad . \tag{11}$$

and the total cost is

$$C_{\min} \approx C_{se}N_R \log N_T + M_{D0}\frac{\ell+1}{\ell}(C_e N_T + C_d N_T \log N_T). \tag{12}$$

The above equation depends on the important parameters for scaling of multitarget tracking, as previously argued, except that it does not depend on combinations of $\overline{R}_0$ with $\alpha\tau$ because of the approximation in Eq. (4b).

VI. Analysis of the General Algorithm

Let $\xi \equiv \alpha\tau/(2R_0\mathcal{M}_D)$, where the symbol for the number of TDS is now $\mathcal{M}_D$ to make a distinction for the limit of Eq. (4b). Instead of taking the approximation in Eq. (6a) leading to Eq. (6b), use the binomial expansion and $N_G = \rho V_G$ to obtain from Eq. (6a):

$$N_G = \rho\frac{\gamma(\ell)}{\ell+1}(R_0)^{\ell}\sum_{i=1}^{\ell+1}\binom{\ell+1}{i}\xi^{i-1}. \tag{13}$$

To find C_{min} it is useful to find the partial of Eq. (13) with respect to $\mathcal{M}_D$:

$$\partial N_G/\partial\mathcal{M}_D = -\rho\frac{\gamma(\ell)}{\ell+1}\frac{(R_0)^{\ell}}{\mathcal{M}_D}\sum_{i=1}^{\ell} i\binom{\ell+1}{i+1}\xi^{i}. \tag{14}$$

This allows us to obtain an equation for the number of TDS that minimize the cost of Eq. (9):

$$\mathcal{M}_{D0}^{\ell+1} = M_{D0}^{\ell+1}\sum_{i=1}^{\ell}\frac{i}{\ell}\binom{\ell+1}{i+1}(\frac{1}{\xi})^{\ell-i}. \tag{15}$$

The above is a polynomial equation for $\mathcal{M}_{D0}$ of ℓ terms and of degree $\ell+1$. For $l=1$, $\mathcal{M}_{D0} = M_{D0}$.

We now verify the C_{min} of Eq. (12) (evaluated when $\alpha\tau >> 2\overline{R}_0$) is an overestimate when it is not true that $\alpha\tau >> 2\overline{R}_0$. We compare $\mathcal{M}_{D0}$ and M_{D0} for a given N_T and N_R, and with a fixed estimate of $\overline{\rho V}_{IG}$ through $\overline{\rho V}_G(\overline{R}_0)$. Let τ_L be some value of a scan length for which $\alpha\tau_L >> 2\overline{R}_0$ and for which Eq. (10) was evaluated to be $M_{D0}(\tau_L)$.

Then, using the ratio of Eq. (15) at τ and at τ_L and using Eq. (10) for $\mathcal{M}_{D0}$ at τ_L, the value of $\mathcal{M}_{D0}$ at some arbitrary scan length is:

$$\mathcal{M}_{D0}^{\ell+1} \approx (\frac{\tau}{\tau_L})^{\ell}[M_{D0}(\tau_L)]^{\ell+1}\sum_{i=1}^{\ell}\frac{i}{\ell}\binom{\ell+1}{i+1}(\frac{2\overline{R}_0\mathcal{M}_D}{\alpha\tau})^{\ell-i}, \tag{16}$$

where the magnitude in the approximation, due to $\mathcal{M}_{D0}(\tau_L) = M_{D0}(\tau_L)$ is given by Eq. (15). The equation above shows how to calculate $\mathcal{M}_{D0}$ for an arbitrary scan length given that it is evaluated for τ_L. Also, Eqs. (9), (13) and (16) give the cost of the gating process in terms of relevant parameters N_T, N_R, ρ, τ, R_0, (and therefore $\overline{R}_0$) and combinations of them. Notice that each of the terms in Eqs. (13) and (16) have their contribution in τ in the form of τ^i, where i is some positive integer. Thus N_G (and $\overline{N}_G$) and $\mathcal{M}_{D0}^{\ell+1}$ decrease as τ decreases on some interval $(0, \tau_L)$. Notice also that as $\overline{N}_G$ and $\mathcal{M}_{D0}$ decrease, the cost as given by Eq. (9) decreases. And since for $\tau = \tau_L$, $\mathcal{M}_{D0}$ $= M_{D0}(1+\text{higher orders in } 2\overline{R}_0/\alpha\tau_L)$, the large scan length case cost is an overestimate of the cost for the general case with a smaller scan length and with the other parameters held fixed.

VII. Discussion

Our choice of the time and dynamics dependent component of R_G in Eq. (3) might not be the only choice possible in that other models might exist which allow a range search. However, making multiple projections of the data to be searched seems to be an option for consideration whenever the bounds on the search range are large due to large timestamp differences. Related to this is that not all of the search algorithms require that the coordinate ranges per search be identical in all directions. It might be possible to separate, by directional components, both R_0 and the time dependent component of R_G. What is required is that one search range per coordinate - and per report when doing a search on tracks - be given to the search algorithm such that all pairs which meet Eq. (1) are satisfied. If it is not true that on the average $\rho V_G(R_0) \approx qN$, where N is the number of tracks when doing a search on tracks and where $V_G(R_0)$ is determined as in our paper, it might be profitable to separate $V_G(R_0)$ into search components or to calculate $V_G(R_0)$ by a different prescription. However, it should be noted that Eqs (10) and (12) are independent of R_0 and so the previous suggestion might not improve execution speed in the large scanlength case. On the other hand, execution speed in the large scanlangth case might be noticably improved by a search space model that will allow one to seperate $V_G(\delta T)$. Note that an algorithm that uses velocity information may reduce the search volume to a half-sphere or perhaps even a cone or a rectangle. This has the effects of changing $\gamma(\ell)$ but leaving Eq. (12), and therefore scaling, as is.

In addition to the above considerations, the value of the optimal number for M_D depends on the computer used, and relative software component speeds, through the constants in Eq. (9). For example a vector computer with good floating point hardware will no doubt reduce $C_{4'}$ over a generic computer; but because C_1 can be reduced by an even larger factor since vectorization of the integration of the equations of motion will be straightforward and possibly complete. This will incline one to have a larger M_D. Performance discussion of different hardware/software setup can get quite detailed and possibly in-house experimentation will be necessary for determining an optimal M_D. Finally, it should be noted that Eq. (9) was not presented with the obvious constraint that M_D be an integer greater than zero. Rounding up to an integer is advisable in implementation when using Eqs. (10) or (16).

VIII. Conclusion

In this paper we have presented an efficient approach to gating in multiple-target tracking. It is general in the sense that it will perform with good scaling even for the case of unequal observation timestamps. An analysis has been provided detailing how the algorithm scales as a function of the important parameters: N_T, N_R, τ, ρ and an estimate of the time independent gate, $V_G(R_0)$. The algorithms has a worst case scaling, in the timestamp parameter, as follows:

$$C_{\min} = O(N_R + M_{D0} N_T \log N_T),$$

where $M_{D0} \propto (\rho N_R \tau^{\ell} / N_T(c + \log N_T))^{\frac{1}{\ell+1}}$. It is also shown how to calculate an optimum number of search data structure in general, and how the algorithm scales in general.

Appendix I : Derivation of Search Radius Covariances and Score Threshold Contribution

Given a report at observation time t and a set of tracks which have been projected to time t, the gating process provides the subset of tracks whose scores exceed a threshold value, denoted here as $S_{\min}$. The successful application of efficient search algorithms to improve scaling with the number of objects requires that a search region (or search radius R_0) be defined independent of the particular track when doing a search upon the track database. A satisfactory expression for the search radius, for example, must ensure that all acceptable tracks fall within a distance R_0 of the report. This appendix derives such an equation for the search region.

Consider a given report and track pair whose respective labels are i and j, whose M-dimensional measurement-prediction difference is denoted $\delta\mathbf{X} \equiv \delta\mathbf{X}_{ij}$ in Eq. (1), and whose position displacement is denoted by $\delta\mathbf{R} \equiv \delta\mathbf{R}_{ij}$. To obtain a search criterion, use the axiom that if the score $S_{ij} \geq S_{\min}$, then the maximum possible score S_0 (given $\delta\mathbf{X}$) must also exceed $S_{\min}$. For this value of $\delta\mathbf{X}$, therefore, we compute the covariance matrix Γ_0 which maximizes S_{ij}. The matrix Γ_0 depends directly on $\delta\mathbf{X}$, and, therefore, so does S_0. Comparison of S_0 with $S_{\min}$ establishes a necessary condition between $\delta\mathbf{X}$ and $S_{\min}$ for all satisfactory track-report pairs. This condition gives an upper bound on the search area or radius, but not necessarily a least upper bound. For each report, one can identify a subset of tracks which should contain all tracks meeting the scoring criteria as well as a small number which do not. Performing this identification for all reports results in a set of candidate track-report pairs. Computing the actual scores of these pairs and selecting those whose scores exceed the threshold $S_{\min}$ completes the gating process.

We begin with Eq. (1) and notice that both the argument of the exponential in the numerator and the determinant of the the covariance matrix Γ in the denominator are invariant under similarity transformations of the coordinates. Assume that a transformation exists which renders Γ diagonal with nonzero elements given by the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_M$. Here d is the dimension of the (Euclidean) report state vector space. Then for a given report-track pair, the score is

$$-\log S \;=\; \frac{1}{2}\sum_{k=1}^{d}\frac{\delta X_k^2}{\lambda_k} \;+\; \frac{d}{2}\log 2\pi \;+\; \frac{1}{2}\sum_{k=1}^{d}\log\lambda_i \quad . \tag{A1}$$

Note that the vector δXhas also undergone transformation, although we retain the same notation for convenience. For a given value of $\delta\mathbf{X}$, we now wish to find the set of eigenvalues $\{\lambda_k\}$ which maximize the score. This requires that we take the partial derivative of the right hand side of Eq. (A1) with respect to each eigenvalue λ_k and set the resulting expression equal to zero. In this way we obtain the condition

$$\lambda_k = \delta X_k^2, \;\; \forall k \quad . \tag{A2}$$

Substituting this in Eq. (A1) gives us

$$-\log S_0 = \frac{d}{2}\Big(1 + \log 2\pi + \frac{1}{m}\sum \log \delta X_k^2\Big) \quad . \tag{A3}$$

From Eq. (A3) and the necessary condition for a given track-report pair to have a score exceeding S_{min}, we obtain the following condition for the search volume (region):

$$\langle \delta X_k^2 \rangle = \frac{1}{2\pi \mathrm{e}} S_0^{-\frac{2}{d}} \le \frac{1}{2\pi \mathrm{e}}(S_{\text{min}})^{-\frac{2}{d}} \quad . \tag{A4}$$

$$\langle \delta X_k^2 \rangle = \Big(\prod_{k=1}^{d} \delta X_k^2\Big)^{\frac{1}{d}} \tag{A5}$$

is the geometric average of the components of the difference $\delta\mathbf{R}$ between the track and report positions in the basis which diagonalizes the covariance matrix.

Equations (A4-5) constitute our most general result. This case, is not straightforward in allowing the definition of a search radius, since the geometric average and arithmetic average of the squared components of the vector $\delta\mathbf{X}$ satisfy

$$\langle \delta X_k^2 \rangle \le \frac{|\delta \mathbf{X}|^2}{d} \tag{A6}$$

by a theorem in mathematical analysis. One can see the reason by considering the circumstance in which the sensors have isotropic resolution. Then the covariance matrices will have d-fold degenerate eigenvalues so that

$$\lambda_k = \frac{1}{d}|\delta\mathbf{X}|^2 \quad \text{(isotropic case)} \tag{A7}$$

will maximize the score in Eq. (A1) and the result corresponding to Eq. (A4) is

$$|\delta \mathbf{R}|^2 \leq |\delta \mathbf{X}|^2 \leq \frac{d}{2\pi e}(S_{\min})^{-\frac{2}{d}} \equiv R_0^2 \qquad (A8)$$

Because the magnitude of the track-report displacement appears on the left-hand side (LHS), the expression on the right-hand side (RHS) constitutes a search radius, denoted as R_0.

For the more general case, it is necessary to have an estimate of the limits of anisotropy of the set of residual covariances for the coordinate system where the near neighbors search is done. Overestimates can be obtained if the set of eigenvalues for all the residual covariances are known. Obtaining the eigenvalues is an $O(N_T)$ operation which for two and three dimensions does not have to be expensive, as we will suggest latter. (this should be a note.) For the jth track in the database, let the ratio of the kth principal axis to the longest principal axis be given by ${\alpha_k}^j$, where ${\alpha_k}^j \leq 1$. The eigenvalues of matrix Γ_j should then satisfy

$$\lambda_k^j = {\alpha_k^j}^2 \lambda^j \qquad (A9)$$

where λ^j is the largest eigenvalue of Γ_j. We now choose the Γ_j which has the smallest geometric mean of the $\alpha^2{}'s$. For this Γ, then, with Eq. (A9) and Eq. (A1) and computing the optimum value of λ gives us

$$\lambda = \frac{1}{d}\sum\Big(\frac{\delta X_k}{\alpha_k}\Big)^2 \qquad (A10)$$

$$S_0^{-\frac{2}{d}} = 2\pi e\Big(\prod \alpha_k^2\Big)^{\frac{1}{d}}\frac{1}{d}\sum\Big(\frac{\delta X_k}{\alpha_k}\Big)^2 \ . \qquad (A11)$$

Denoting the geometric average as before, noting the necessary condition on S_0, and remembering that $\alpha_k \leq 1$, we obtain

$$|\delta \mathbf{R}|^2 \leq |\delta \mathbf{X}|^2 \leq \sum\Big(\frac{\delta X_k}{\alpha_k}\Big)^2 \leq \frac{d}{2\pi e\langle\alpha_k^2\rangle}(S_{\min})^{-\frac{2}{d}} \equiv R_0^2 \ . \qquad (A12)$$

Again since $|\delta \mathbf{R}|^2$ appears on the LHS, the extreme RHS defines a search radius R_0.

Appendix II : Scenario Description

For the sake of completeness we state that the basic scenario presented in this paper consisted of $N_T = N_R$ reports and tracks obtained by adding Gaussian noise to ground truth trajectories. The ground truth was in an XYZ common coordinate system. The trajectories were Newtonian and parabolic with a constant gravitational force in the Z-direction; also they were determined by choosing initial positions and final destinations random in X, Y with $Z = 0$ and bounded on the $X - Y$ plane with one rectangle for the initial positions and another for the final destinations. The two rectangles were set sufficiently far apart so that the average flight would be about 25 minutes in duration with the ground truth objects flying at about 11 km/sec. A scan consisted of a sampling on the trajectories in the form of reports with timestamps within a scan of length τ. The only parameters changed to generate all the data presented in the plots were N_R (and N_T), τ ,M_D (the number of track data structure copies), and the object density as a consequence of changing N_R (and N_T).

References

[1] Nagarajan,V., Chidambara, M.R., and Sharma, R.M., " New approach to improved detection and tracking in track-while-scan radars,Part 2: Detection, track initiation, and association." *IEE Proceedings, Vol. 134*, PT. F, No. 1, Feb. 1987.

[2] Blackman, Samuel S., Multiple-Target Tracking with Radar Applications., Artech House, Dedham,MA,1986.

[3] "SDI tracker/correlator design document (third ed.)," Ball Systems Engineering Division and Daniel H. Wagner Associates, Delivered to U.S. NRL April 15, 1989.

[4] Jain, A.K., and Dubes, R.C., Algorithms for Clustering Data.Prentice Hall, Englewood Cliffs, N.J.,1988.

[5] Uhlmann, J.K., "Enhancing multidimensional tree structures using a bi-linear decomposition," NRL Memorandum Report (1990).

[6] Uhlmann,J.K., " A remedy for the hierarchical divisibility problem using a bilinear decomposition," Berkeley Research Associates Report No. BRA-89-W040R, Berkeley, CA, June 1989.

[7] Bentley, Jon Louis, "Multidimensional binary trees for associative searching." *Communications of the ACM*, 18(9): 509-517, Sept. 1975.

[8] Friedman, J.H., Baskett, F.B., and Shusteck, L.J., "Finding nearest neighbors," *IEEE Trans. Comput.*, Oct. 1975.

[9] Friedman,J.H.,Bentley, J.L., and Finkel, R.A., " An algorithm for finding best matches in logarithmic expected time," *ACM Trans. Math Software*, vol. 3, Sept. 1977.

[10] Samet, Hanan, The Design And Analysis of Spatial Data Structures. Addison-Wesley Publishing Co., New York, N.Y.,1990.

[11] Mehlhorn, Kurt, Multi-Dimensional Searching and Computational Geometry. Springer-Verlag, Berlin, 1984.

[12] Sedgewick, Robert, Algorithms. Second Edition, Addison-Wesley, New York, N.Y., 1988.

Acknowledgements We would like to thank Ellis Hyman for his editorial guidance; Gopal

Patnaik for suggesting the use of a contour plot instead of a surface plot for Fig. 1 ; and Felix Rosenthal for his help with the graphics.

This work was performed under US N.R.L. contract No. N00014-86-C-2197

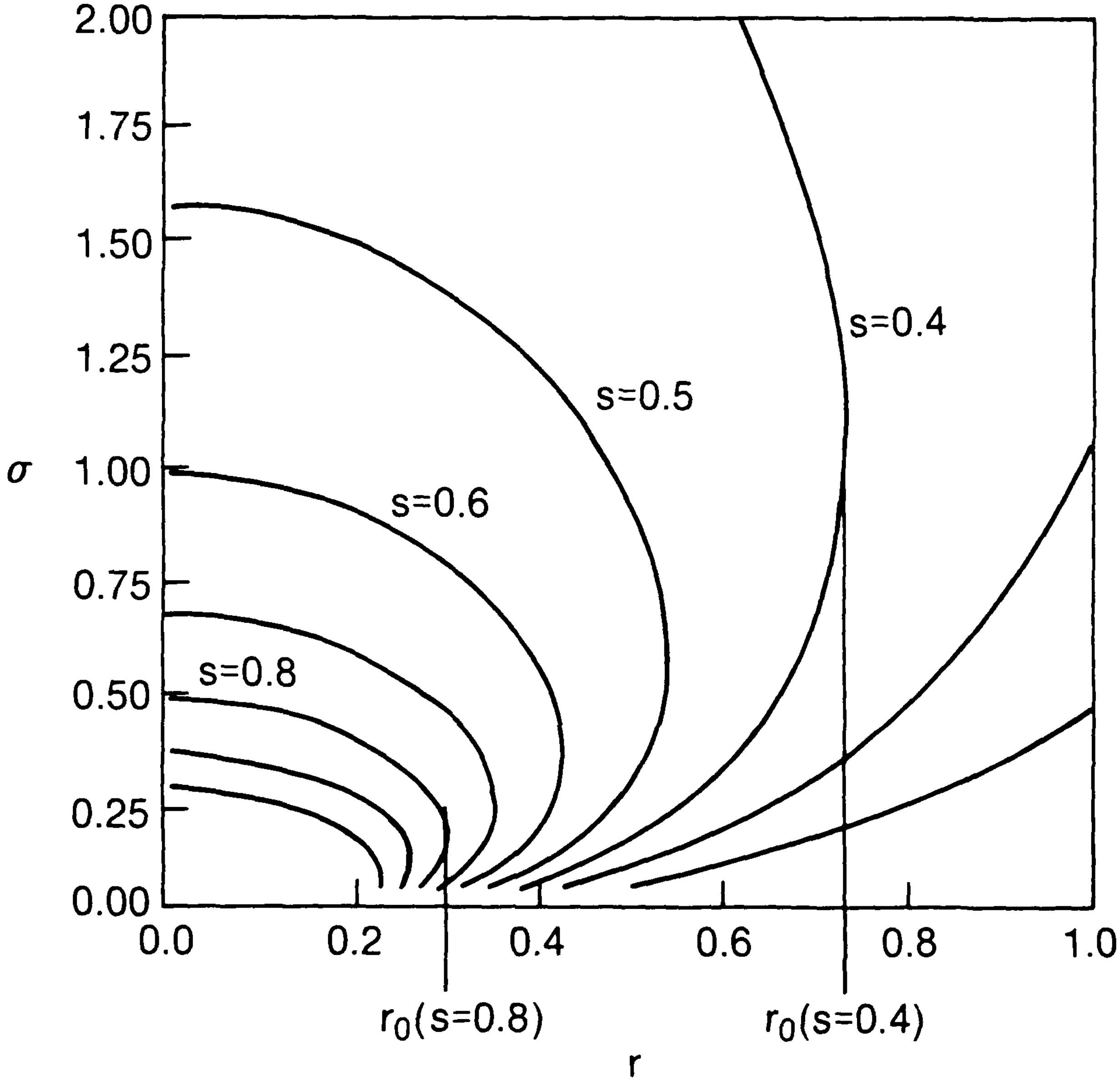


Fig. 1 — Contour plot of the score function $s = e^{-r^2/2\sigma}/\sqrt{2\pi\sigma}$. The r_0's are the chosen suitable time independent radius component due to covariances and score threshold contributions.

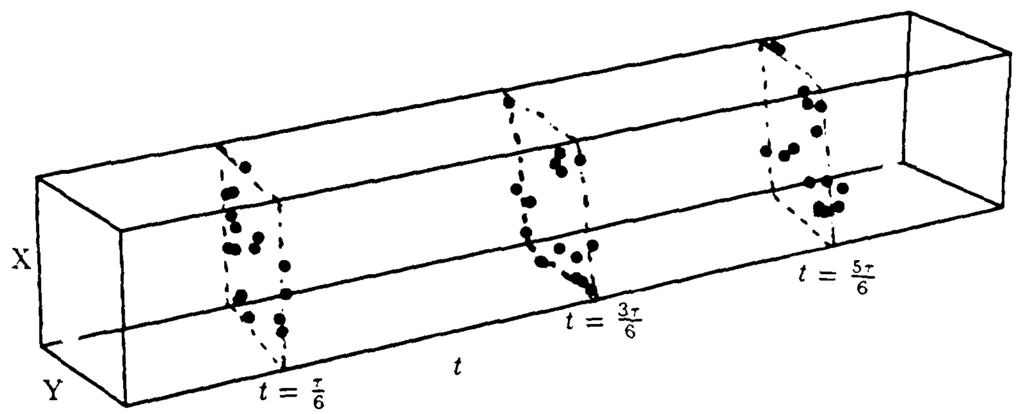
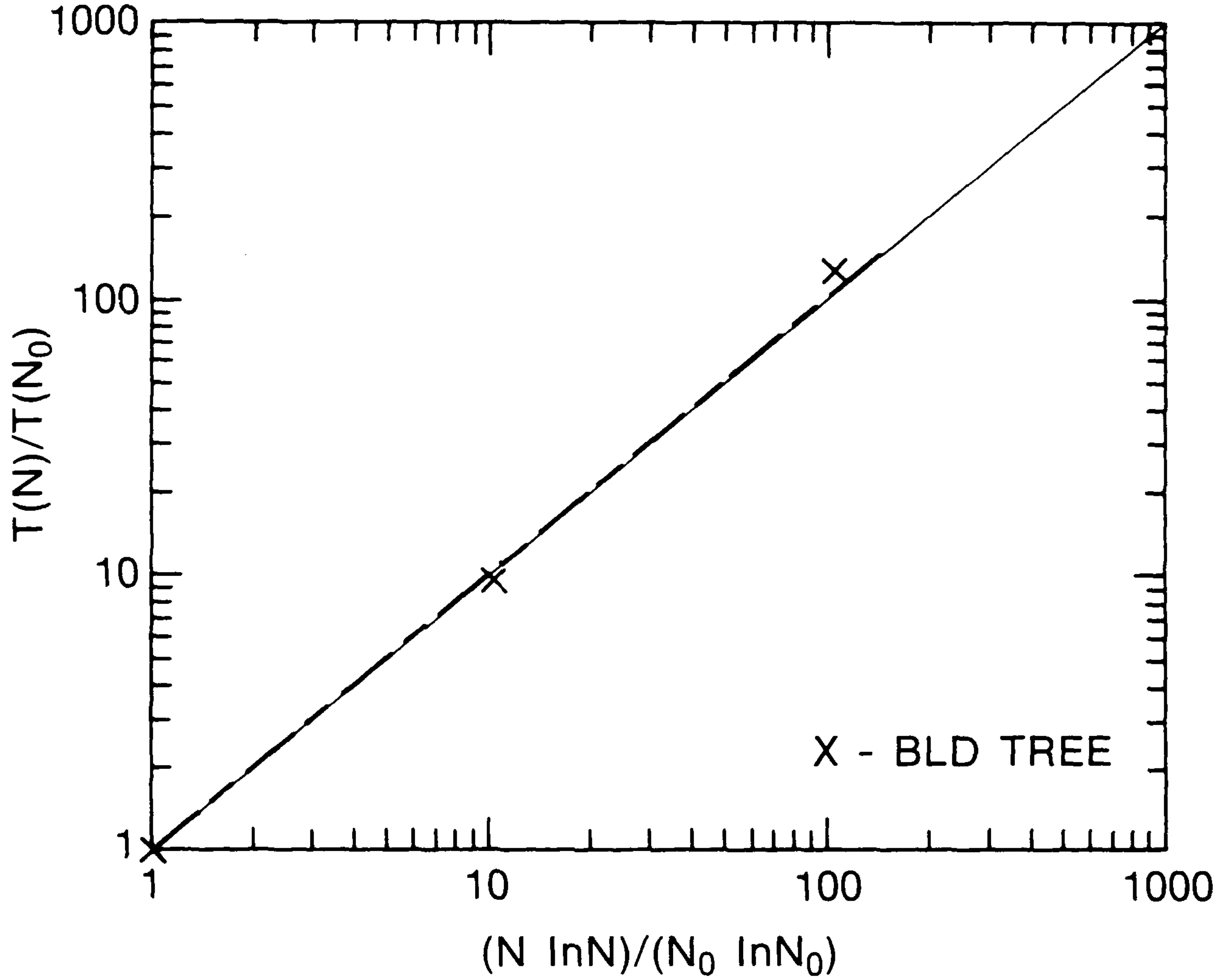
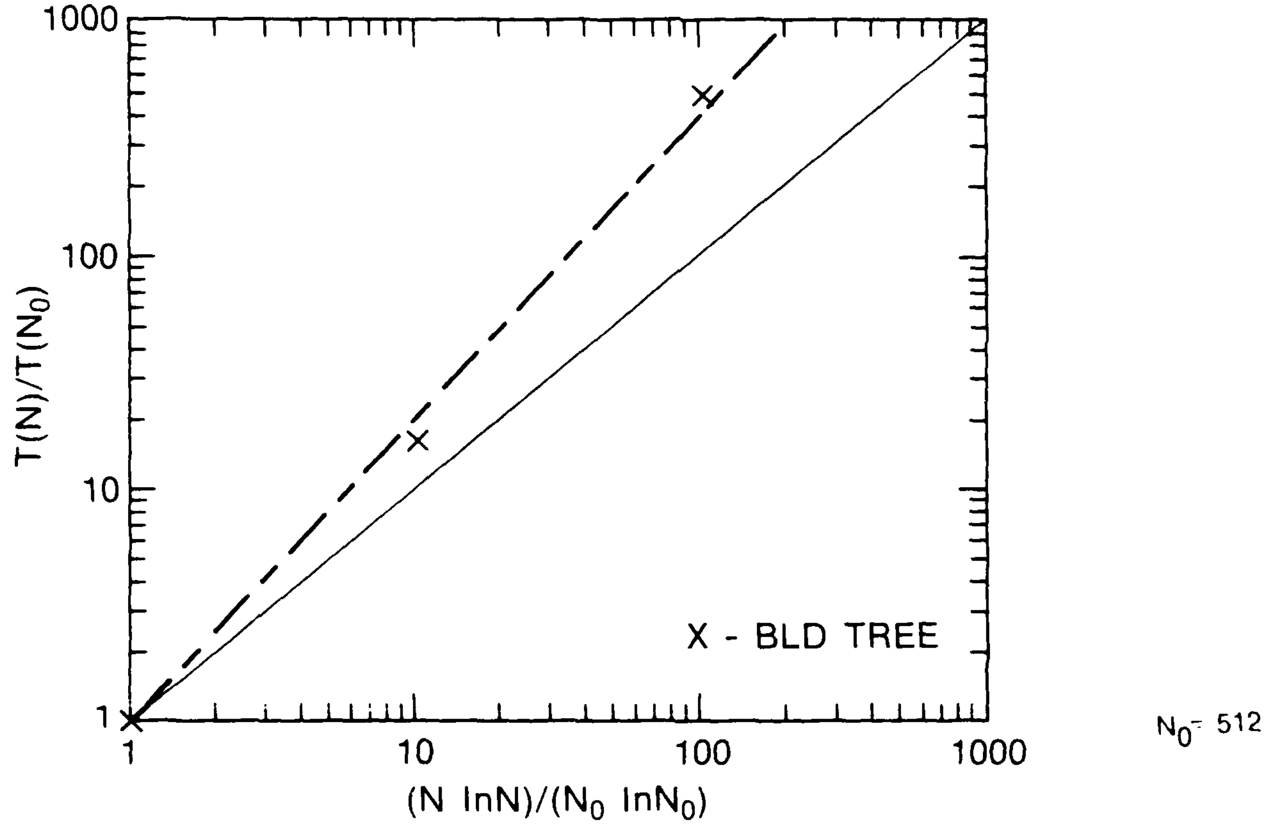


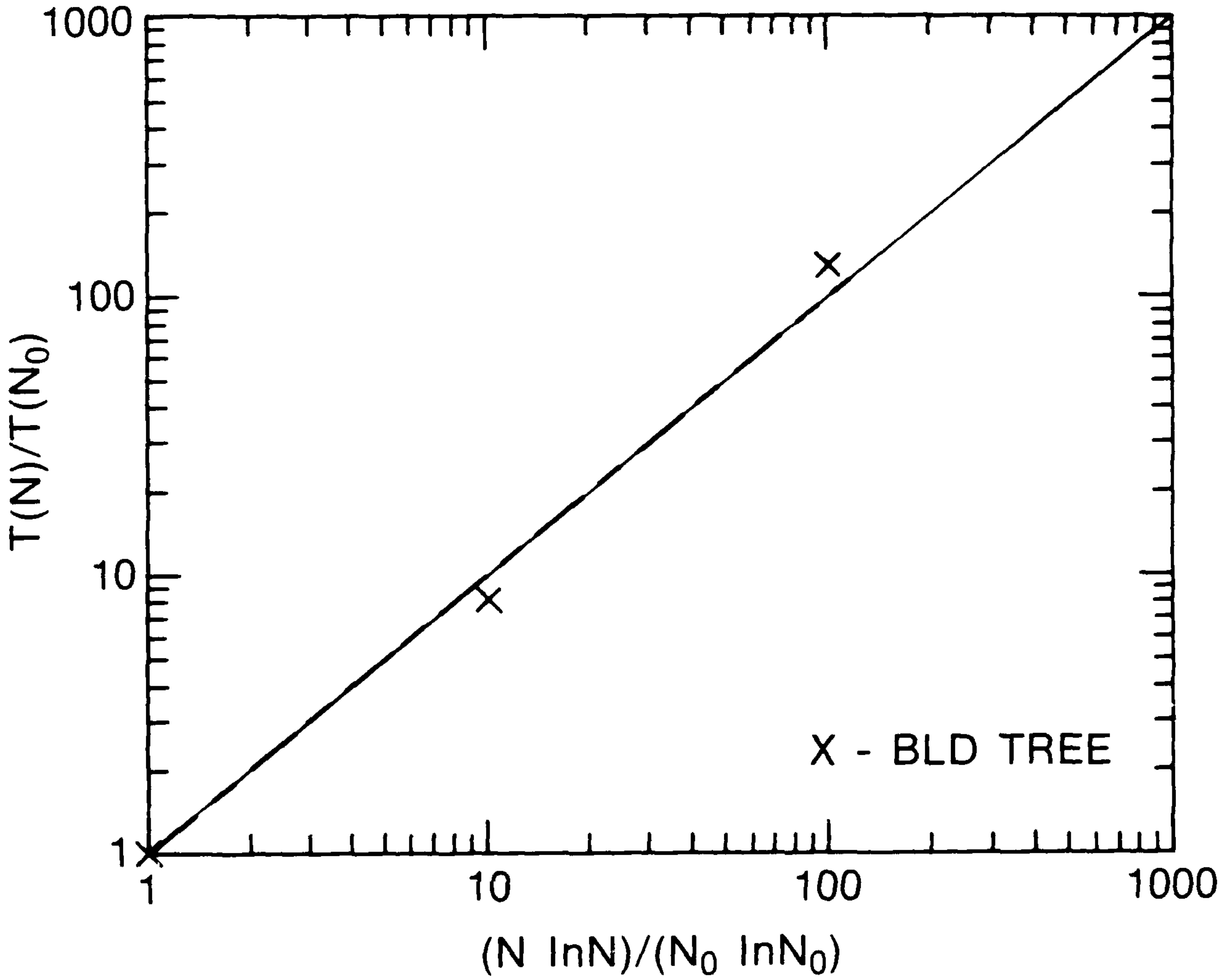
Fig. 2 — Illustration of the time spacing of the multiply projected track data structures for $M_D = 3$ and a scan of length τ. The data structures are projected to a time $t_i = (2i - 1)\tau/2M_D$, where $i = 1, \ldots, M_D$.



Plot 1 — Search time scaling of search algorithm. Search for neighbors of N reports among data strictures with N tracks; return Pairs.



Plot 2 — Search time scaling. T = increased to 10 sec; M_D = 1.



Plot 3 — Search time scaling. Multiple track data copies. TS = 10 sec; M_D = 1 copy for N = 512; 5 copies for N = 4K and 32K.